WHAT IS CLAIMED IS:

1.   A device capable of usage in a storage system including an initiator, a primary target, and a secondary target, the device being capable of configuration for asynchronous remote mirroring functionality and comprising:

a target controller capable of communicating with the initiator and performing operations requested by the initiator; and

a process on the target controller capable of receiving a stream of command requests from the initiator, performing the requested commands, and asynchronously relaying the requested commands to a secondary target while ensuring precedence graph equivalence between received and relayed commands, the process embedding task attributes in the relayed commands capable of increasing concurrency in commands performed by the secondary target.

2.   The device according to Claim 1 further comprising:

a process on the target controller that receives a command stream from the initiator that specifies task attributes describing an execution sequence order, and reconveys the commands and task attributes to the secondary target so that the execution order is the same in the primary target and the secondary target.

3.   The device according to Claim 1 further comprising:

a process on the target controller that receives a command stream from the initiator that omits specification of task attributes describing an execution sequence order, determines implied task attributes from the command stream based on ordinality of tasks, and conveys the commands and implied task attributes to the secondary target, enabling improvement in concurrency in command execution by the secondary target.

4.   The device according to Claim 3 further comprising:

a process on the target controller that determines implied task attributes from the

command stream further comprising:

a process that infers an intended precedence graph of a primary initator by

tracking concurrency of issued tasks; and

a process that conveys the inferred preference graph to the secondary

target.

5.   The device according to Claim 1 further comprising:

a process on the target controller that determines implied task attributes from the

command stream further comprising:

a process that responds to a transition from less than two outstanding write

tasks to multiple outstanding write tasks by marking the multiple

outstanding write tasks as Simple tasks.   \

6.   The device according to Claim 1 further comprising:

a process on the target controller that determines implied task attributes from the

command stream further comprising:

a process that responds to a transition from multiple outstanding write

tasks to no outstanding write tasks by marking a subsequent write

task as an Ordered task.

7.   The device according to Claim 1 further comprising:

a process on the target controller that determines implied task attributes from the

command stream further comprising:

a process that responds to a transition from multiple outstanding write

tasks to no outstanding write tasks by inserting an Ordered no_op

command and tentatively marking a subsequent write task as an

Ordered task, subject to demotion to a Simple task on subsequent

multiple outstanding tasks.

8.   The device according to Claim 1 further comprising:

a process on the target controller that determines implied task attributes from the

command stream further comprising:

a process that tracks an active number of write tasks for a nexus relating

the initiator, targets, and logical units (LUNs);

a process responsive to no active write tasks that inserts a no-op command

marked as Ordered in a precedence graph to be propagated to the

secondary target;

a process responsive to no active write tasks that sets a no-op_inserted flag

to zero;

a process responsive to no active write tasks and a no-op_inserted flag set

to one that marks a new write task temporarily as an Ordered and

promoted task;

a process responsive to no active write tasks and a no-op_inserted flag set

to zero that marks a new write task as an Ordered task;

a process responsive to one active write task in which a promoted task

exists that demotes the promoted task to a Simple task; and

a process responsive to at least one active write task that marks a new

write task to a Simple task.

9.   The device according to Claim 1 further comprising:

a process on the target controller that determines command attributes to ensure

correct operation at the secondary target in the absence of the initiator

supplying the attributes. ·

10.   The device according to Claim 1 wherein:

the target controller is a Small Computer Systems Interface (SCSI) device and the

task attributes are SCSI task attributes including Simple, Ordered, Head of

Queue, and Auto Contingent Allegiance (ACA) task attributes.

11.   A method of ordering commands in a communication system comprising:

receiving a stream of commands;

performing the commands;

embedding task attributes in the relayed commands to increase concurrency in

      commands performed by the target; and

relaying the commands to a target.

12.   A method according to Claim 11 further comprising:

inferring an intended precedence of the requested commands; and

embedding task attributes in the relayed commands according to the intended

      precedence to increase concurrency in commands performed by the target.

13.   A method according to Claim 11 further comprising:

receiving a command stream that omits specification of task attributes describing

      an execution sequence order;

determining implied task attributes from the command stream; and

conveying the commands and implied task attributes to enable improvement in

      concurrency in commands performed by the target.

14.   A method according to Claim 11 further comprising:

receiving a command stream from the initiator that specifies task attributes

      describing an execution sequence order; and

reconveying the commands and task attributes to the secondary target so that the

      execution order is the same in the primary target and the secondary target.

15.   A method according to Claim 11 further comprising:

tracking an active number of write tasks for a nexus relating the initiator, targets,

      and logical units;

responding to no active write tasks by inserting a no-op command marked as

      Ordered in a precedence graph to be propagated to the secondary target;

responding to no active write tasks by setting a no-op_inserted flag to zero;

responding to no active write tasks and a no-op_inserted flag set to one by

      marking a new write task temporarily as an Ordered and promoted task;

responding to no active write tasks and a no-op_inserted flag set to zero by
marking a new write task as an Ordered task;

responding to one active write task in which a promoted task exists by demoting
the promoted task to a Simple task; and

responding to at least one active write task by marking a new write task to a
Simple task.

16.   The method according to Claim 11 further comprising:

determining command attributes to ensure correct operation in the absence of the
initiator supplying the attributes.

17.   The method according to Claim 11 further comprising:

tracking concurrency of issued tasks;

inferring an intended precedence graph based on the tracked tasks; and

applying Ordered and Simple task attributes to a stream of pipelined commands to
convey a precise execution sequence order for issued commands according
to the inferred precedence graph to improve command execution
efficiency.

18.   The method according to Claim 11 wherein:

operating in a Small Computer Systems Interface (SCSI) device; and

the task attributes are SCSI task attributes including Simple, Ordered, Head of
Queue, and Auto Contingent Allegiance (ACA) task attributes.

19.   The method according to Claim 11 further comprising:

relaying commands in a remote asynchronous mirroring application.

20.   A method of ordering commands in a communication system comprising:

communicating information via an internet Small Computer Systems Interface
(iSCSI) transport protocol; and

embedding Small Computer Systems Interface (SCSI) task attributes in a received
command stream to ensure an ordering behavior sufficient for attaining
concurrency performance and correct algorithm operation.

21.   A method of ordering commands in a communication system comprising:

receiving a stream of command requests;

embedding task attributes in the relayed commands to improve transaction

    ordering for a remote mirroring application using an ordered transport; and

relaying the requested commands.

22.   An article of manufacture comprising:

a controller usable medium having a computable readable program code embodied

    therein for ordering commands in a communication system, the

    computable readable program code further comprising:

    a code capable of causing the controller to receive a stream of command

        requests;

    a code capable of causing the controller to perform the requested

        commands;

    a code capable of causing the controller to infer an intended precedence of

        the requested commands;

    a computable readable program code capable of causing the controller to

        embed task attributes in the relayed commands according to the

        intended precedence to increase concurrency in commands

        performed by the target; and

    a code capable of causing the controller to relay the requested commands

        to a target.

23.   An article of manufacture comprising:

a controller usable medium having a computable readable program code embodied

    therein for ordering commands in a communication system, the

    computable readable program code further comprising:

    a code capable of causing the controller to communicate information via

        an internet Small Computer Systems Interface (iSCSI) transport

        protocol; and

    a code capable of causing the controller to embed Small Computer

        Systems Interface (SCSI) task attributes in a received command

stream to ensure an ordering behavior sufficient for attaining

concurrency performance and correct algorithm operation.

24.   An article of manufacture comprising:

a controller usable medium having a computable readable program code embodied

therein for ordering commands in a communication system, the

computable readable program code further comprising:

a code capable of causing the controller to receive a stream of command

requests;

a code capable of causing the controller to relay the requested commands;

and

a code capable of causing the controller to embed task attributes in the

relayed commands to improve transaction ordering for a remote

mirroring application using an ordered transport.

25.   A device in a communication system comprising:

means for receiving a stream of command requests;

means for performing the requested commands;

means for embedding task attributes in the relayed commands capable of

increasing concurrency in commands performed by the target; and

means for relaying the requested commands to a target.

26.   A remote mirroring method comprising:

communicating information between a primary initiator and a secondary target;

inferring a task precedence graph intended by the primary initiator by tracking

concurrency of issued tasks; and

conveying the inferred task precedence graph to the secondary target.

27.   The remote mirroring method according to Claim 26 further comprising:

sensing the intended task precedence graph by tracking transition points

dependent on ordinality of commands in a command stream.

28.   The remote mirroring method according to Claim 27 wherein tracking transition points further comprises:

  switching at a first transition point from a multiple outstanding write task

    condition, even briefly, to a no outstanding write task condition; and

  switching at a second transition point from one or fewer outstanding write tasks to

    multiple outstanding write tasks.

29.   The remote mirroring method according to Claim 28 further comprising:
inferring the intended task precedence graph comprising:

  marking multiple outstanding tasks at the second transition point as Simple

    tasks;

  marking a first task following the first transition point as an Ordered no-op

    command; and

  marking as Ordered all commands issued by the initiator in a sequence.

30.   The remote mirroring method according to Claim 29 further comprising:
after marking the no-op command:

  tentatively marking a first task following the first transition point as

    Ordered, wherein the marked Ordered task is demoted to a Simple

    task when the number of outstanding commands exceeds one.